



用肺部 X-光片與症狀做 COVID-19 預測

B09705039 資管三 劉惟恩

B09705044 資管三 王裕勳

小阿姨我不想努力ㄌ



大綱



涵蓋主題

1. Opening
2. ML Model 1: Symptom Prediction
3. DL Model 2: X_ray Prediction
4. Combination of models and Insights

Opening

Idea and Motivation

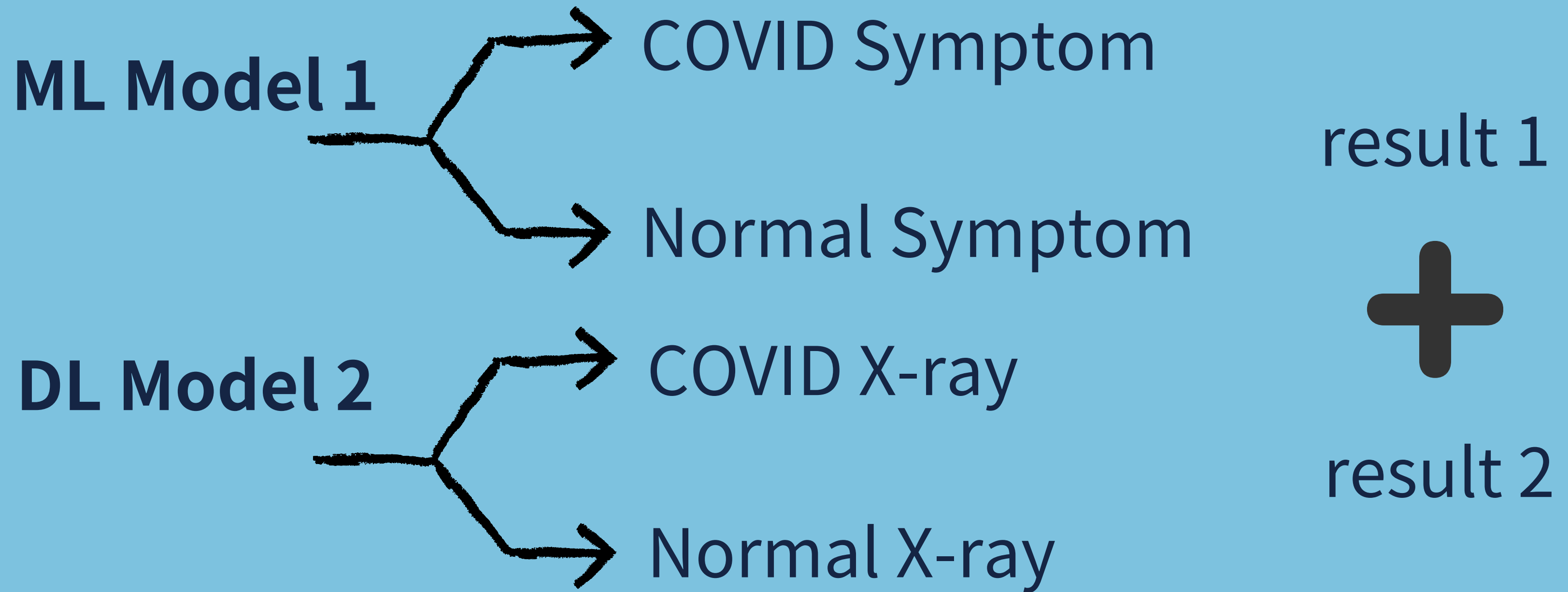


發想題目動機

1. 除了提高快篩的準確度外，我們的目標是避免重症患者被誤判的可能性，給予更準確的醫療資源。
2. 查詢網路後發現目前的模型停留於指使用其中一種資料做預測有可能會有盲點（如：醫療影像 or 症狀預測）。

=> 於是想到也許可以結合兩種模型的預測優點來達到更好的準確度

研究框架





ML Model 1: Symptom Prediction

COVID / Normal



Data Analysis / Feature Transformation

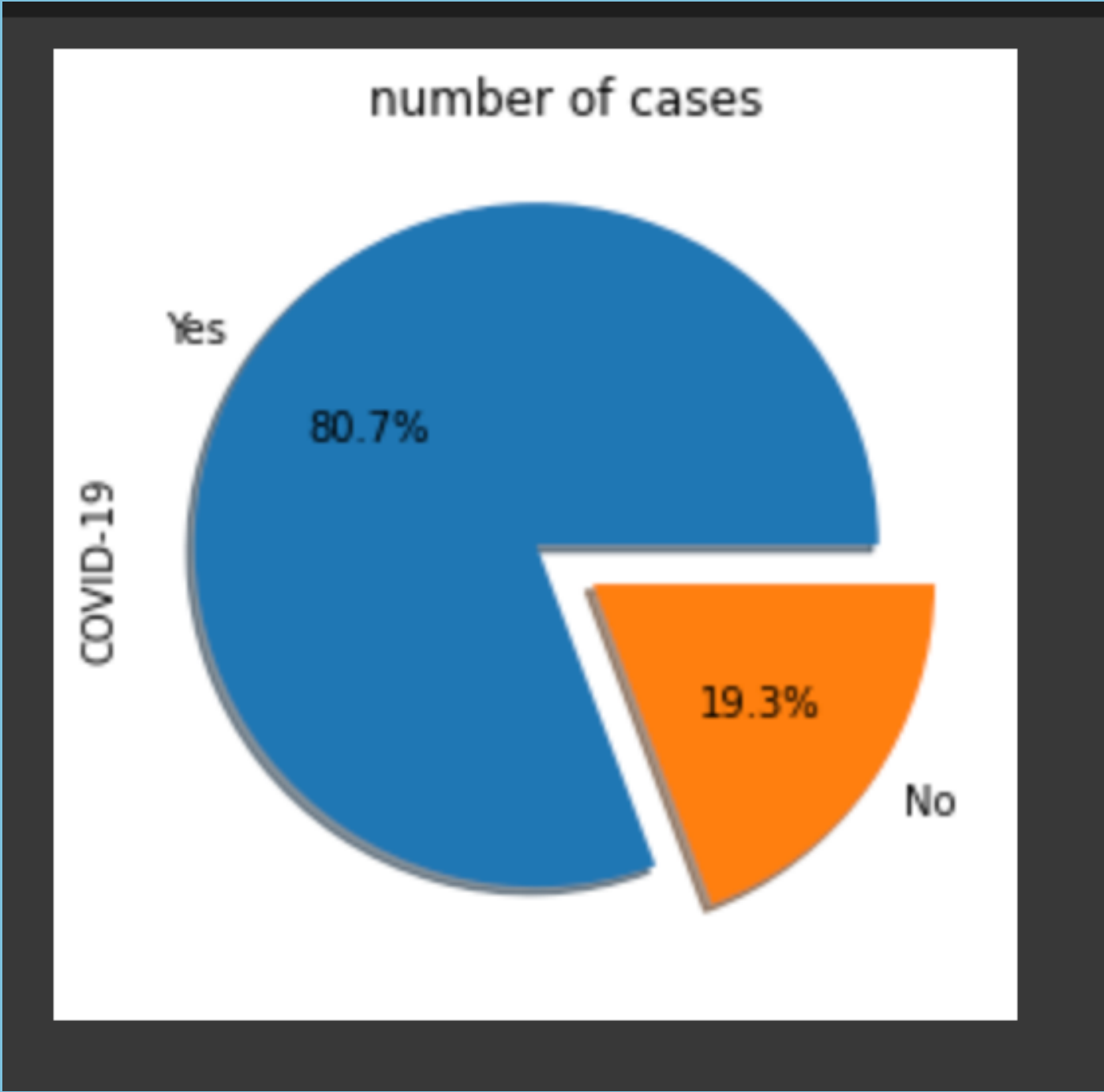
Dataset : [Kaggle Symptoms and COVID Presence \(May 2020 data\)](#).

表中 Yes / No 代表該患者有沒有該症狀，共統計 5434 名不同患者、20 種特徵

Breathing	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	Headache
Yes	Yes	Yes	Yes	Yes	No	No	No
Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	No	No	Yes	No	No
Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Yes	Yes	Yes	No	No	No	No	No
Yes	Yes	Yes	No	No	No	Yes	No
Yes	Yes	Yes	No	Yes	Yes	No	No
Yes	Yes	Yes	No	Yes	No	Yes	No
Yes	Yes	Yes	No	No	Yes	No	No
Yes	Yes	Yes	No	No	No	Yes	No
Yes	Yes	Yes	Yes	Yes	Yes	No	No
Yes	Yes	Yes	Yes	No	Yes	Yes	No
Yes	Yes	Yes	Yes	No	Yes	No	No
??	??	??	??	??	??	??	??

- 1.經過分析後發現沒有遺漏值需要填補，而確診者與未確診的比例如右下圖
- 2.將資料集裡的 Yes / No 分別改為 1 / 0 的 encoding

#	Column	Non-Null Count		Dtype
0	Breathing Problem	5434	non-null	object
1	Fever	5434	non-null	object
2	Dry Cough	5434	non-null	object
3	Sore throat	5434	non-null	object
4	Running Nose	5434	non-null	object
5	Asthma	5434	non-null	object
6	Chronic Lung Disease	5434	non-null	object
7	Headache	5434	non-null	object
8	Heart Disease	5434	non-null	object
9	Diabetes	5434	non-null	object
10	Hyper Tension	5434	non-null	object
11	Fatigue	5434	non-null	object
12	Gastrointestinal	5434	non-null	object
13	Abroad travel	5434	non-null	object
14	Contact with COVID Patient	5434	non-null	object
15	Attended Large Gathering	5434	non-null	object
16	Visited Public Exposed Places	5434	non-null	object
17	Family working in Public Exposed Places	5434	non-null	object
18	Wearing Masks	5434	non-null	object
19	Sanitization from Market	5434	non-null	object
20	COVID-19	5434	non-null	object



Feature Selection

1. Drop features that have all same values (Wearing Masks, Sanitization from Market).
2. Draw correlation heat map. => Most have low correlation.

	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	Headache	Heart Disease	Diabetes	Hyper Tension	Fatigue	Gastrointestinal	Abroad travel	with COVID Patient
Breathing Problem	1.000000	0.089903	0.159562	0.303768	0.055190	0.075318	-0.098291	-0.062172	-0.073366	0.055427	0.045256	0.000561	-0.075390	0.117795	0.214634
Fever	0.089903	1.000000	0.127580	0.322235	0.081758	0.073953	-0.025160	-0.035416	-0.031462	0.050286	0.079001	-0.060458	-0.008067	0.128726	0.164704
Dry Cough	0.159562	0.127580	1.000000	0.213907	-0.030763	0.086843	-0.043664	-0.035912	0.047566	-0.006593	0.081989	-0.039909	0.008251	0.331418	0.128330
Sore throat	0.303768	0.322235	0.213907	1.000000	0.039450	0.081377	-0.050440	-0.015971	0.002177	0.001938	0.042811	-0.023290	0.025886	0.205986	0.189251
Running Nose	0.055190	0.081758	-0.030763	0.039450	1.000000	-0.022763	-0.014376	0.068479	-0.056750	0.042961	-0.020445	0.007026	-0.014673	0.034526	0.003776
Asthma	0.075318	0.073953	0.086843	0.081377	-0.022763	1.000000	-0.033771	0.037064	0.076783	-0.012060	0.017707	0.006564	0.101909	0.068286	0.005046
Chronic Lung Disease	-0.098291	-0.025160	-0.043664	-0.050440	-0.014376	-0.033771	1.000000	-0.050480	-0.039860	0.046789	-0.010331	-0.047655	-0.050333	-0.088854	-0.062482
Headache	-0.062172	-0.035416	-0.035912	-0.015971	0.068479	0.037064	-0.050480	1.000000	0.048471	0.032390	-0.207489	0.052035	0.097778	0.043589	-0.082101
Heart Disease	-0.073366	-0.031462	0.047566	0.002177	-0.056750	0.076783	-0.039860	0.048471	1.000000	-0.032956	0.049139	-0.058925	0.004121	-0.020761	-0.025593
Diabetes	0.055427	0.050286	-0.006593	0.001938	0.042961	-0.012060	0.046789	0.032390	-0.032956	1.000000	0.042543	-0.043903	0.040651	0.039013	-0.085696
Hyper Tension	0.045256	0.079001	0.081989	0.042811	-0.020445	0.017707	-0.010331	-0.207489	0.049139	0.042543	1.000000	-0.027605	-0.067972	-0.016382	0.027307
Fatigue	0.000561	-0.060458	-0.039909	-0.023290	0.007026	0.006564	-0.047655	0.052035	-0.058925	-0.043903	-0.027605	1.000000	0.009356	-0.068401	-0.027383
Gastrointestinal	-0.075390	-0.008067	0.008251	0.025886	-0.014673	0.101909	-0.050333	0.097778	0.004121	0.040651	-0.067972	0.009356	1.000000	0.099577	0.025277
Abroad travel	0.117795	0.128726	0.331418	0.205986	0.034526	0.068286	-0.088854	0.043589	-0.020761	0.039013	-0.016382	-0.068401	0.099577	1.000000	0.080210
Contact with COVID Patient	0.214634	0.164704	0.128330	0.189251	0.003776	0.005046	-0.062482	-0.082101	-0.025593	-0.085696	0.027307	-0.027383	0.025277	0.080210	1.000000

Feature Selection (Con't)

3. Split data into training and testing data with 8 : 2 and then split training into training and validation data with 8:2.
4. Using select K best (K = 15) with p_value by univariate linear regression test on training data. (We tried all kinds of K and found K = 15 has the best results.)

Feature Selection Result:

['Breathing Problem', 'Fever', 'Dry Cough', 'Sore throat', 'Asthma',
'Chronic Lung Disease', 'Headache', 'Heart Disease', 'Diabetes',
'Hyper Tension', 'Abroad travel', 'Contact with COVID Patient',
'Attended Large Gathering', 'Visited Public Exposed Places',
'Family working in Public Exposed Places']

Machine Learning Algorithms

We tried the following algorithms with hyper parameter tuning:

1. Logistic Regression
2. Random Forest Classifier
3. Gradient Boosting Classifier
4. K Neighbors Classifier
5. Decision Tree Classifier
6. Support Vector Classifier

Results

Below are the Accuracy, F1-Score, Recall and Precision of all the ML models with the best hyper parameter tuning.

	Model	Accuracy
0	Support Vector Machines	0.983441
3	Random Forest	0.982521
4	Decision Tree Classifier	0.982521
5	Gradient Boosting Classifier	0.982521
1	KNN	0.981601
2	Logistic Regression	0.964121

	Model	F1_Score
0	Support Vector Machines	0.989643
3	Random Forest	0.989036
4	Decision Tree Classifier	0.989036
5	Gradient Boosting Classifier	0.989036
1	KNN	0.988479
2	Logistic Regression	0.977470

Results (con't)

	Model	Precision
3	Random Forest	0.994200
4	Decision Tree Classifier	0.994200
5	Gradient Boosting Classifier	0.994200
0	Support Vector Machines	0.991926
1	KNN	0.991908
2	Logistic Regression	0.983721


	Model	Recall
0	Support Vector Machines	0.987371
1	KNN	0.985075
3	Random Forest	0.983927
4	Decision Tree Classifier	0.983927
5	Gradient Boosting Classifier	0.983927
2	Logistic Regression	0.971297

Results (con't)

1. To avoid patients infected with covid-19 not diagnosed by the model, we picked the model with high recall as the first priority.
 2. Then we pick the model with high f1-score since it is the combination of Precision and Recall.
 3. Finally we pick the model with high accuracy score.
- => Except logistic regression, the other five models all did great in this dataset especially SVC, decision tree and random forest.

Conclusion

All models did quite well prediction on the testing dataset, so we can conclude that this is a great model for covid-19 prediction.



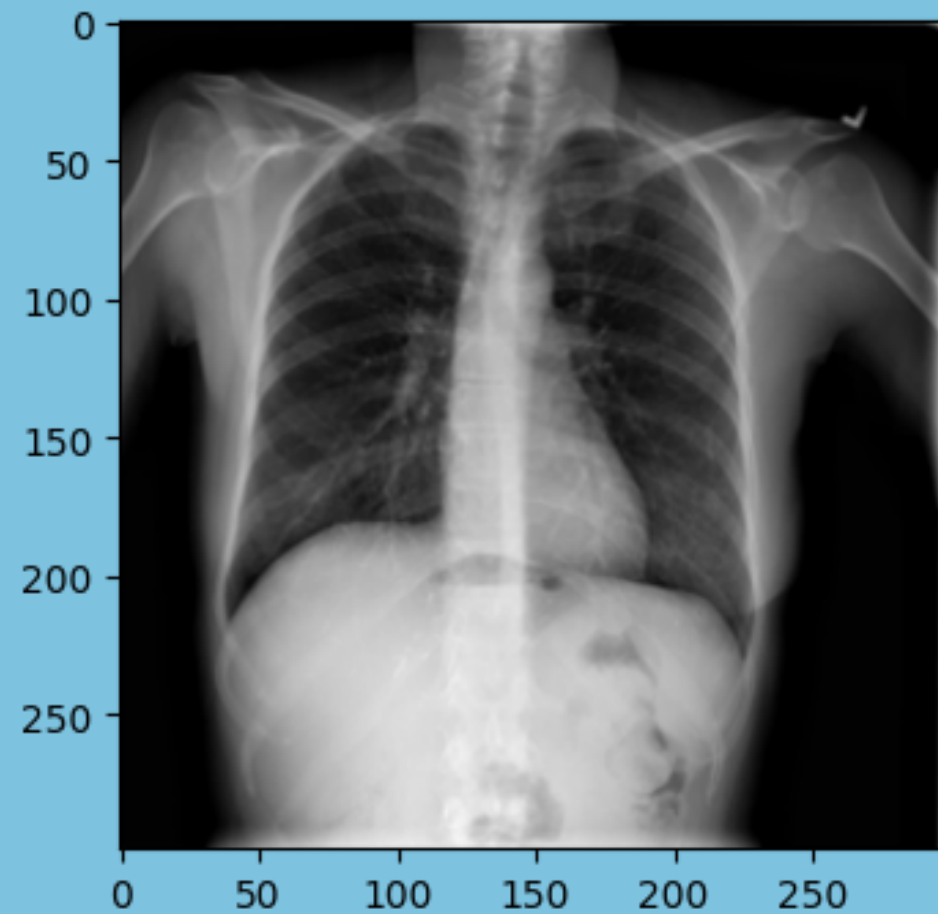
DL Model 2: X-ray Prediction

COVID / Normal

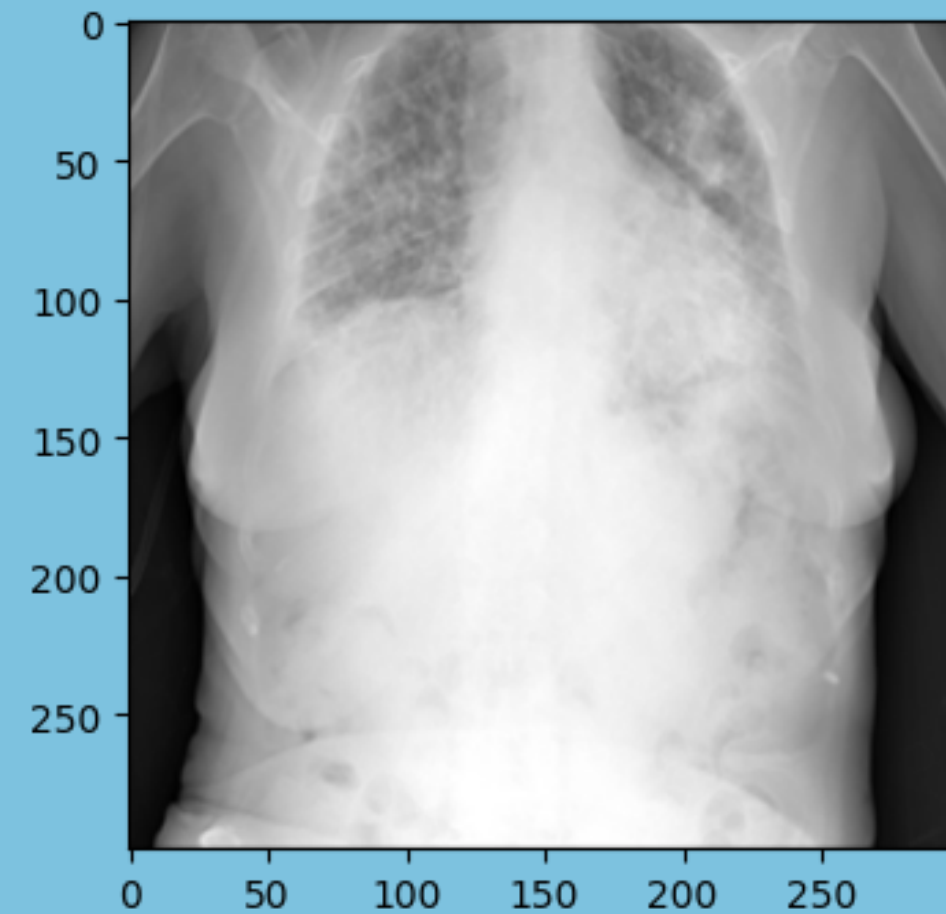


資料搜集

1. 從kaggle上找已有的資料集
2. 資料為胸腔的x ray圖，採用的lable為Covid(確診)和Normal(沒確診)



Normal



Covid

資料前處理

1. 取7232(3616normal+3616covid)筆來跑
2. 將照片轉成 numpy，大小為(100,100,3(rgb))
3. resize 成相同大小

```
from skimage import color
normal_images = []
for i in normal_images_dataset[:5000]:
    img = cv2.imread(str(i))
    img = cv2.resize(img, (100, 100))
    normal_images.append(img)
    # normal_images.append(color.rgb2gray(img))
normal_images = np.array(normal_images)
normal_images.shape
```

Normalization

使用StandardScaler來normalization，以x_train來fit

```
for i in range(x_train_value.shape[1]):  
    for j in range(x_train_value.shape[2]):  
        for k in range(x_train_value.shape[3]):  
            x = np.array([x_train_value[:,i,j,k]]).T  
            scaler1 = StandardScaler().fit(x)  
            x_n = scaler1.transform(x)  
            x_train_value[:,i,j,k] = x_n.reshape(x.shape[0])  
            y = np.array([x_test_value[:,i,j,k]]).T  
            y_n = scaler1.transform(y)  
            x_test_value[:,i,j,k] = y_n.reshape(y.shape[0])
```

Model

使用CNN，由於資料量不大，使用三層卷積層加一層Linear

```
class MyModule(nn.Module):
    def __init__(self, input_dim):
        super(MyModule, self).__init__()
        self.cnn = nn.Sequential(
            nn.Conv2d(input_dim, 64, 3, 1, 1), # [64, 100, 100]
            nn.BatchNorm2d(64),
            nn.ReLU(),
            nn.MaxPool2d(2, 2, 0), # [64, 50, 50]
            nn.Dropout2d(p=0.2),
            nn.Conv2d(64, 128, 3, 1, 1), # [64, 50, 50]
            nn.BatchNorm2d(128),
            nn.ReLU(),
            nn.MaxPool2d(2, 2, 0),
            nn.Dropout2d(p=0.2), # [64, 25, 25]
            nn.Conv2d(128, 256, 3, 1, 1), # [64, 25, 25]
            nn.BatchNorm2d(256),
            nn.ReLU(),
            nn.MaxPool2d(2, 2, 0), # [64, 12, 12]
            nn.Dropout2d(p=0.2)
        )
        self.fc = nn.Sequential(
            nn.Linear(128*25*25, 512),
            nn.ReLU(),
            # nn.Dropout(0.6),
            nn.Linear(512, 1)
        )

    def forward(self, x):
        out = self.cnn(x)
        out = out.view(out.size()[0], -1)

        return self.fc(out)
```

Train

1. 使用adam做為optimizer
2. 使用MSELoss做為criterion
3. 使用CosineDecay來動態調整Learning rate
4. 共跑50個epoch

Predict

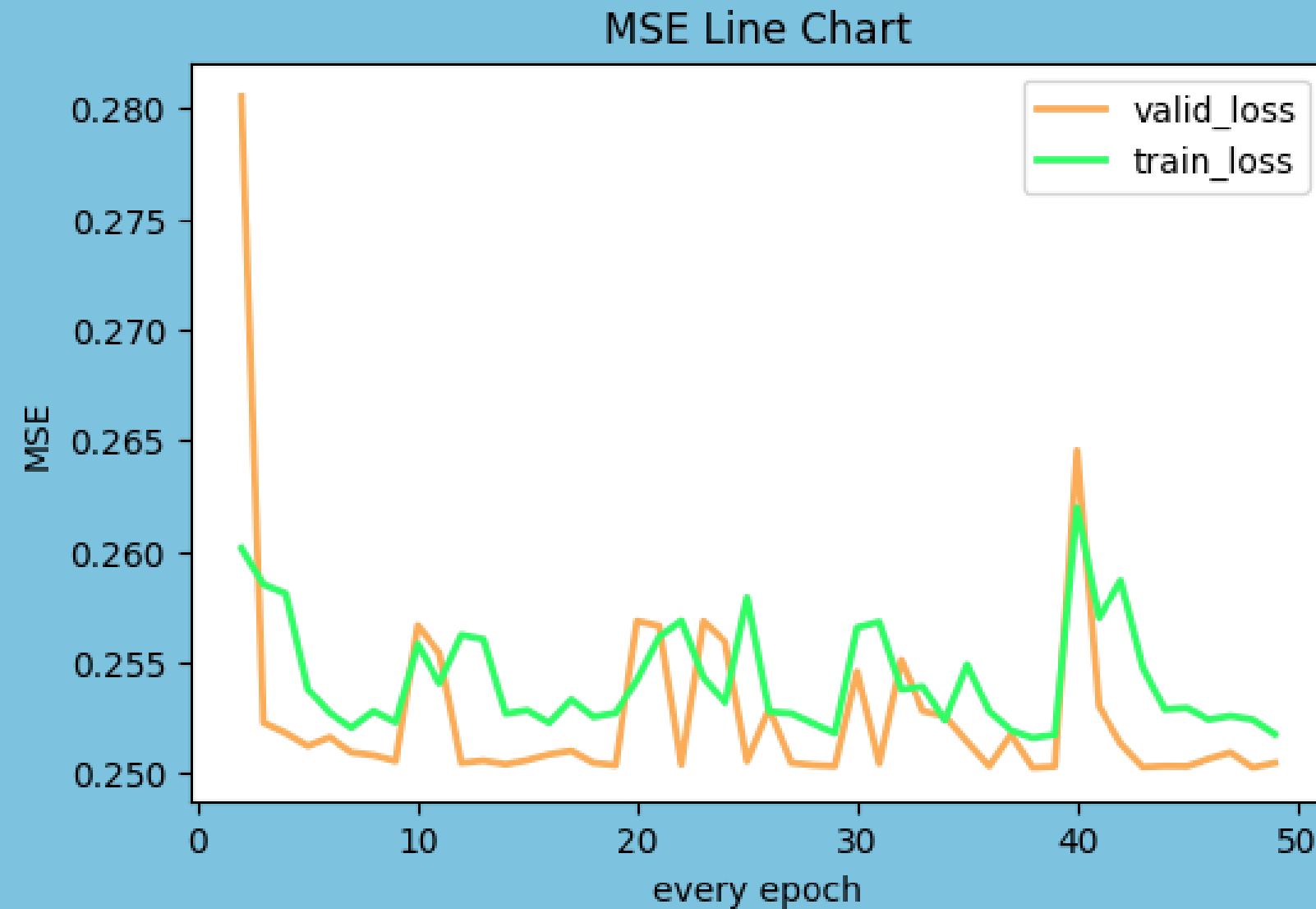
F-1 Score: 0.5938480853735092

Accuracy: 0.5525587828492393

Recall: 0.6709219858156028

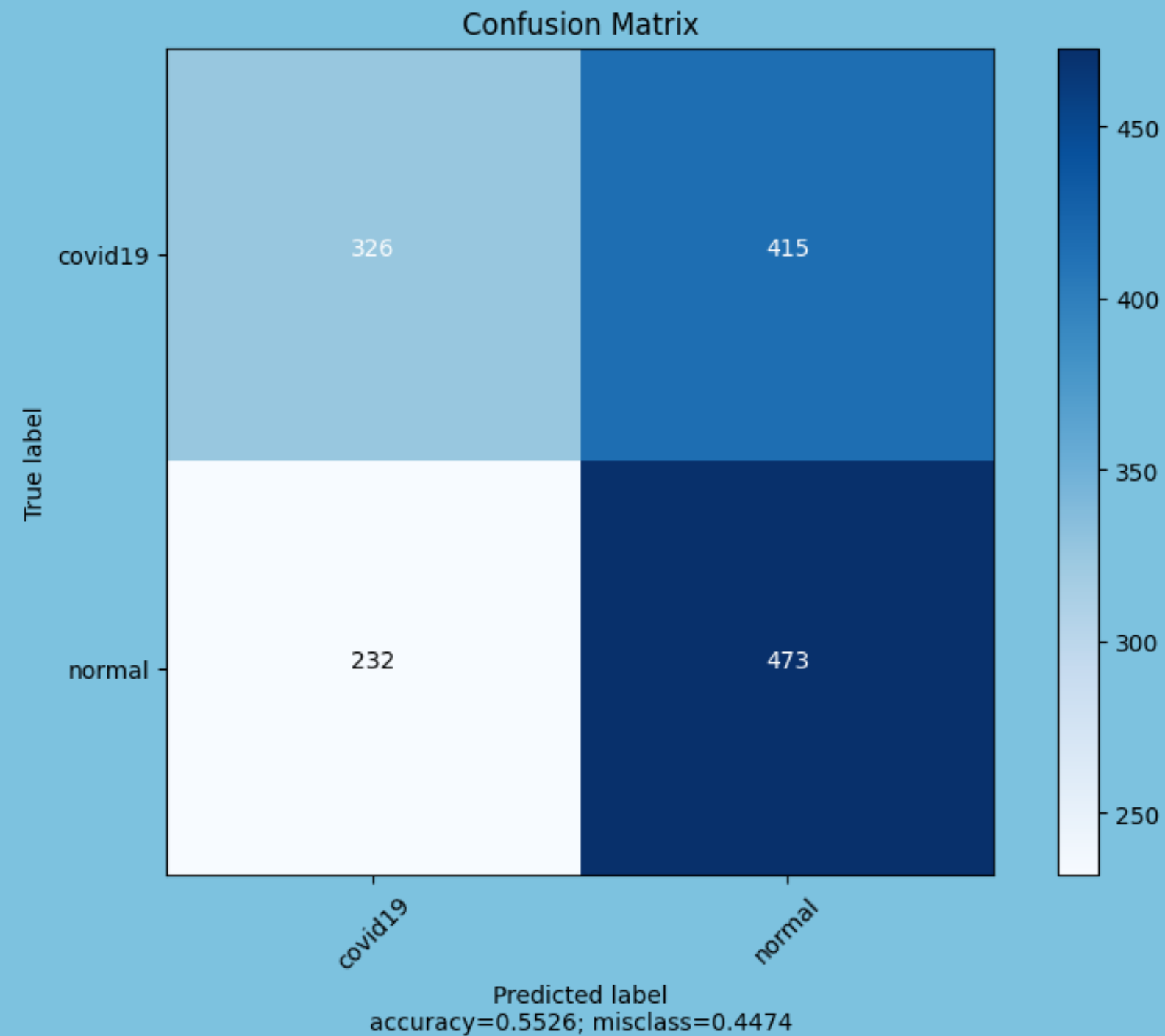
Precision 0.5326576576576577

Visualization



Train loss and Valid loss gradually descend

Visualization(Cont)



Conclusion

The accuracy is only about 0.55, and as the confusion matrix shows, the consequent of the model is not as good as the statistic analysis, because our goal is to avoid every patient who infected covid will be correctly diagnosed, it is obvious an worse model. But whatever how we modify the parameter, the result is not enhance at all, and we also refer to another code which use the same dataset and CNN, the result is not better than our, so we conclude that maybe CNN is not a good choice to run this dataset.



Combination of models and Insights

COVID / Normal



How to combine two models?

Since both models are classification models we considered a efficient to combine it:

- 1.If the first and the second model predicts that the person has covid-19 we have great confidence that the person is infected with covid-19 and may already have very severe symptoms which should remain in hospital under observation.
- 2.If either model predicts that the person has covid-19 we may consider them as mild symptoms and tell them to be aware of there selfs at home.

How to combine two models? (Cont)

3. If neither of them shows evidence of a person is infected with covid-19, we can say that the person is not infected with covid-19.

This method is more efficient and careful when diagnosing whether a patient is infected with covid-19. By using this method we can have higher recall and much better accuracy compared with only using one of the model to predict.



Future exploration

COVID / Normal



What else can we do?

1. We only have two datasets separated so it is hard to conclude which combination method is better. However, since these medical data are confidential patent informations it is hard to collect more data for better prediction models.
2. The result of classification using CNN is not as good as what we expected, according to the result we obtain, it seems that the CNN model we build is too simple to this problem, we should try another more complex model.



**Thanks For
Listening !**

